IN THE UNITED STATES PATENT AND TRADEMARK OFFICE

Patent Application

Applicant(s): Parida et al.
Docket No.: YOR920030299US1
Serial No.: 10/661,322
Filing Date: September 12, 2003
Group: 2129
Examiner: Peter D. Coughlan

Title: Discovering Permutation Patterns

APPEAL BRIEF

Mail Stop Appeal Brief - Patents
Commissioner for Patents
P.O. Box 1450
Alexandria, VA 22313-1450

Sir:

Applicants hereby appeal the final rejection dated September 5, 2006, of claims 1 through 29 of the above-identified patent application.

REAL PARTY IN INTEREST

The present application is assigned to International Business Machines Corporation, as evidenced by an assignment recorded on March 2, 2004 in the United States Patent and Trademark Office at Reel 014391, Frame 0140. The assignee, International Business Machines Corporation, is the real party in interest.

RELATED APPEALS AND INTERFERENCES

There are no related appeals and interferences.

STATUS OF CLAIMS

Claims 1 through 29 are presently pending in the above-identified patent application. Claims 1-29 are rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter. Claims 1-17, 20-26, and 29 are rejected under 35 U.S.C. §102(b) as being anticipated by Floratos, "DELPHI: A Pattern-based Method for Detecting Sequence Similarity," claims 18 and 27 are rejected under 35 U.S.C. §103(a) as being unpatentable over Floratos, in view of Savitch, "Problem Solving with C++," and claims 19 and 28 are rejected under 35 U.S.C. §103(a) as being unpatentable over Floratos, and Savitch, and further in view of Fredman, "Two Applications of a Probablistic Search Technique: Sorting X+Y and Building Balanced Search Trees."

STATUS OF AMENDMENTS

There have been no amendments filed subsequent to the final rejection.

SUMMARY OF CLAIMED SUBJECT MATTER

Independent claim 1 is directed to a method (FIG. 4: 320) of discovering permutation patterns (FIG. 1) from an input string (FIG. 4: 305) having a plurality of characters, each character being from an alphabet (page 4, lines 6-16; page 8, lines 14-27), the method comprising the steps of: selecting a new portion of the input string (FIG. 4: 305), the new portion differing from a previously selected portion of the input string (FIG. 4: 305) by at least one new character of the input string (FIG. 4: 305; page 9, lines 5-20); determining one or more values for how many of the at least one new characters are in the portion of the input string (FIG. 4: 305; page 9, lines 5-20; page 10, lines 8-25); determining which, if any, names in a plurality of sets of names have changed by selection of the new portion, the plurality of sets comprising a first set and a plurality of additional sets, wherein the first set corresponds to all of the characters in the alphabet and to values of how many of the characters of the alphabet are in the previously selected portion, wherein the values are names for the first set, and wherein each additional set comprises names corresponding to selected pairs of names from a single other set (FIG.

2: steps 110-145; page 9, lines 5-20; page 10, line 12, to page 11, line 2); and using changes in the names to determine the permutation patterns (FIG. 4: 340; FIG. 2: step 150; page 9, lines 5-20).

Independent claim 20 is directed to an apparatus (FIG. 4: 300) for discovering permutation patterns (FIG. 1) from an input string (FIG. 4: 305) having a plurality of characters, each character being from an alphabet (page 4, lines 6-16; page 8, lines 14-27), the apparatus comprising: a memory (FIG. 4: 315); at least one processor (FIG. 4: 310) coupled to the memory, the at least one processor configured: to select a new portion of the input string (FIG. 4: 305), the new portion differing from a previously selected portion of the input string by at least one new character of the input string (FIG. 4: 305; page 9, lines 5-20); to determine one or more values for how many of the at least one new characters are in the portion of the input string (FIG. 4: 305; page 9, lines 5-20; page 10, lines 8-25); to determine which, if any, names in a plurality of sets of names have changed by selection of the new portion, the plurality of sets comprising a first set and a plurality of additional sets, wherein the first set corresponds to all of the characters in the alphabet and to values of how many of the characters of the alphabet are in the previously selected portion, wherein the values are names for the first set, and wherein each additional set comprises names corresponding to selected pairs of names from a single other set (FIG. 2: steps 110-145; page 9, lines 5-20; page 10, line 12, to page 11, line 2); and to use changes in the names to determine the permutation patterns (FIG. 4: 340; FIG 2: step 150; page 9, lines 5-20).

Independent claim 29 is directed to an article of manufacture for discovering permutation patterns (FIG. 1) from an input string (FIG. 4: 305) having a plurality of characters, each character being from an alphabet (page 4, lines 6-16; page 8, lines 14-27), the article of manufacture comprising: a computer readable medium containing one or more programs which when executed implement the steps of: selecting a new portion of the input string (FIG. 4: 305), the new portion differing from a previously selected portion of the input string by at least one new character of the input string (FIG. 4: 305; page 9, lines 5-20); determining one or more values for how many of the at least one new characters are in the portion of the input string (FIG. 4: 305; page 9,

lines 5-20; page 10, lines 8-25); determining which, if any, names in a plurality of sets of names have changed by selection of the new portion, the plurality of sets comprising a first set and a plurality of additional sets, wherein the first set corresponds to all of the characters in the alphabet and to values of how many of the characters of the alphabet are in the previously selected portion, wherein the values are names for the first set, and wherein each additional set comprises names corresponding to selected pairs of names from a single other set (FIG. 2: steps 110-145; page 9, lines 5-20; page 10, line 12, to page 11, line 2); and using changes in the names to determine the permutation patterns (FIG. 4: 340; FIG. 2: step 150; page 9, lines 5-20).

In one exemplary embodiment, the method further comprises the step of determining the plurality of levels through the steps of: determining the first set by determining values of how many of each of the characters of the alphabet are in the previously selected portion; and determining the additional sets by assigning names for a given additional set to selected pairs of names from another of the sets, wherein each assigned name is unique to the names for a selected pair (page 9, lines 5-20; page 10, line 12, to page 11, line 2; page 12, line 15, to page 13, line 2).

In one exemplary embodiment, the method further comprises the step of determining, for a name that has changed in the sets of names, a location in the input string (FIG. 4: 305) that corresponds to the changed name (page 9, lines 5-20).

In one exemplary embodiment, each of the names in the sets of names corresponds to a pattern, and the step of using changes further comprises the step of selecting permutation patterns (FIG. 1) from the patterns (FIG. 4: 340; FIG. 2: step 150; page 9, lines 5-20).

In one exemplary embodiment, the method further comprises the step of comparing names that have changed in the sets of names to a database comprising a plurality of stored names (page 13, line 28, to page 14, line 5).

STATEMENT OF GROUNDS OF REJECTION TO BE REVIEWED ON APPEAL

Claims 1-17, 20-26, and 29 are rejected under 35 U.S.C. §102(b) as being anticipated by Floratos, claims 18 and 27 are rejected under 35 U.S.C. §103(a) as being

unpatentable over Floratos, in view of Savitch, and claims 19 and 28 are rejected under 35 U.S.C. §103(a) as being unpatentable over Floratos, and Savitch, and further in view of Fredman.

ARGUMENT

Section 101 Rejections

Claims 1-29 were rejected under 35 U.S.C. §101 because the claimed invention is directed to non-statutory subject matter. In particular, the Examiner asserts that the invention has not been limited to a substantial practical application. In the final Office Action, the Examiner asserts that discovering permutations has no practical purpose, and that no function or application has been stated for the invention.

The Supreme Court has stated that the "[t]ransformation and reduction of an article 'to a different state or thing' is the clue to patentability of a process claim." *Gottshalk v. Benson*, 409 U.S. 63, 70, 175 U.S.P.Q. (BNA) 676 (1972). In other words, claims that require some kind of transformation of subject matter, which has been held to include intangible subject matter, such as data or signals, that are representative of or constitute physical activity or objects have been held to comply with Section 101. *See, for example, In re Warmerdam,* 31 U.S.P.Q.2d (BNA) 1754, 1759 n.5 (Fed. Cir. 1994) or *In re Schrader*, 22 F.3d 290, 295, 30 U.S.P.Q.2d (BNA) 1455, 1459 n.12 (Fed. Cir. 1994).

Thus, as expressly set forth in each of the independent claims, the claimed methods or system describe discovering permutation patterns from an input string having a plurality of characters, each character being from an alphabet, and transform the input string to permutation patterns. This transformation to permutation patterns provides a useful, concrete and tangible result. For example, the Background section of the present disclosure describes how such permutation patterns are utilized in medical applications related to genes and proteins. Thus, contrary to the Examiner's assertion that no function or application has been stated for the invention, Appellants note that the Background section of the present disclosure describes how such permutation patterns are utilized in medical applications related to genes and proteins (see, page 1, line 12, to page 2, line

20). The final result of the cited claims, i.e., permutation patterns, are useful, concrete and tangible results.

Appellants submit that each of claims 1-29 are in full compliance with 35 U.S.C. §101, and accordingly, respectfully request that the rejection under 35 U.S.C. §101 be withdrawn.

Independent Claims 1, 20 and 29

Independent claims 1, 20, and 29 were rejected under 35 U.S.C §102(b) as being anticipated by Floratos Regarding claim 1, the Examiner asserts that Floratos teaches "using changes in the names to determine the permutation patterns" (page 457, col. 2, lines 5-15; 'Permutation patterns' of applicant are equivalent to 'K' of Floratos). In the final Office Action, the Examiner asserts that

> (Floratos, p 462, C1:5 through C2:4) Floratos explains the number of times that a residue R appears in P, which is the bases of permutation of a pattern. Floratos also illustrates a quantitative result which is the frequency of a specific pattern (or permutation). Another example of permutation is on page 457, C1:17-25 Floratos illustrates a backbone (or permutation) of '100111' which is an example of the string of 'A...DEF.' If Floratos were to 'change' the name to 'AB DEF' then the backbone (or permutation) would be '110111.'
> (Page 17, last paragraph, of the final Office Action.)

Appellants note that Floratos is directed to a different problem than the present disclosure. Floratos is directed to "identifying ***sequence*** *similarity* between a query sequence and a database of proteins." (Page 455, first paragraph; emphasis added.) Floratos searches for an *ordered sequence in a string*. The claims of the present disclosure are directed to *discovering* ***permutation*** *patterns*. As would be apparent to a person of ordinary skill in the art, permutation patterns indicate that the patterns are related to a *non-ordered set of characters*. For instance, dictionary.com teaches that the permutations of (1,2,3) are (1,2,3) (2,3,1) (3,1,2) (3,2,1) (1,3,2) (2,1,3). The examples presented by the Examiner, however, are ***not*** *permutation patterns*, as would be understood by a person of ordinary skill in the art.

For example, the Examiner asserts that Floratos explains the number of times that a residue R appears in P, which is the bases of permutation of a pattern and illustrates a quantitative result which is the frequency of a specific pattern (or

permutation). Appellants note that, in the text cited by the Examiner, Floratos teaches that

> during the search phase, two mechanisms are available for prohibiting the association of two sequences along a potentially low-complexity region. The first permits the use of only a "linguistically rich" subset of the patterns discovered during information gathering. In particular, for each pattern P, we define its variability $v(P)$ as
>
> $$v(P) = \frac{\max_R \{\text{number of times that the residue } R \text{ appears in } P\}}{\text{total number of positions in } P \text{ covered by residues}}$$
>
> and allow the user access to a global parameter V which dictates that a pattern P is employed in the search phase only if $v(P) < V$.
>
> The second mechanism allows the disregarding of local similarities of low informational content
>
> (Page 462, column 1, lines 11-24.)

The Examiner also asserts that another example of permutation is on page 457, col. 1, lines 17-25, where Floratos illustrates a backbone (or permutation) of '100111' which is an example of the string of 'A..DEF.' Appellants note that Floratos teaches that,

> given a pattern P, the *backbone* of P is defined as a string over the alphabet {1, 0} obtained from P by turning every residue of P into the character "1" and every don't-care into the character "0." For example, the backbone of the pattern P = "*A..DFE*" introduced above is the string B = "100111." Backbones partition the set of patterns into equivalent classes, with each class containing all of the patterns sharing the same backbone. A pattern with backbone B is designated as a B-pattern.
>
> (Page 457, column 1, lines 17-25.)

Neither of these citations, however, are examples of permutation patterns, as defined above and as would be understood by a person of ordinary skill in art. Independent claims 1, 20, and 29 require using changes in the names to determine the ***permutation patterns***.

Thus, Floratos does not disclose or suggest using changes in the names to determine the permutation patterns, as required by independent claims 1, 20, and 29.

<u>Additional Cited References</u>

Savitch was also cited by the Examiner for its disclosure of wherein the at least one character is a single character and wherein the step of selecting further comprises selecting a portion of the input string that differs from the previously selected

portion of the input string by moving a window one character, from the previously selected portion, along the input string, the window selecting the new portion of the input string. Appellants note that Savitch is directed to a program using an array. Savitch does ***not*** address the issue of using changes in names to determine permutation patterns.

Thus, Savitch does not disclose or suggest using changes in the names to determine the permutation patterns, as required by independent claims 1, 20, and 29.

Fredman was also cited by the Examiner for its disclosure of wherein the sets of names are stored in a balanced search tree. Appellants note that Fredman is directed to a search method that translates into an insertion sort, and to the construction of probabilistically binary search trees. Fredman, however, does ***not*** address the issue of using changes in names to determine permutation patterns.

Thus, Fredman does not disclose or suggest using changes in the names to determine the permutation patterns, as required by independent claims 1, 20, and 29.

Claims 2 and 21

Claims 2 and 21 are rejected under 35 U.S.C. §102(b) as being anticipated by Floratos. In particular, the Examiner asserts that Floratos discloses to determine the first set by determining values of how many of each of the characters of the alphabet are in the previously selected portion (page 457, col. 1, lines 26-43); and to determine the additional sets by assigning names for a given additional set to selected pairs of names from another of the sets, wherein each assigned name is unique to the names for a selected pair (page 456, col. 2, line 44, to page 457, col. 1, line 7).

In the text cited by the Examiner, Floratos teaches:

> We also define the *offset list* of P with respect to D (or simply the offset list of P, when D is unambiguously implied) as the set of pairs
>
> $L_D(P) == \{(i, j) \mid$ P matches the substring of S_i starting at offset j$\}$.
>
> As an example, consider the pattern P = *"A..DFE"* and the set of sequences
>
> $D = \{S_1 =$ GHASEDFER, S_2 = LKERAHPDFE, S_3 = LKMNAKLD$\}$.
>
> In this set, the pattern P has support 2 (the boldface substrings indicate the sequence regions matching P), and its offset list is $L_D(P) = \{(1, 3), (2, 5)\}$

(Page 456, col. 2, line 44, to page 457, col. 1, line 7.)

Appellants could find no disclosure or suggestion in Floratos of determining the additional sets by assigning names for a given additional set to selected pairs of names from another of the sets, wherein each assigned name is unique to the names for a selected pair. Claims 2 and 21 require the step of determining the plurality of levels through the steps of: determining the first set by determining values of how many of each of the characters of the alphabet are in the previously selected portion; and determining the additional sets by assigning names for a given additional set to selected pairs of names from another of the sets, wherein each assigned name is unique to the names for a selected pair.

Thus, Floratos, Savitch, and Fredman, alone or in any combination, do not disclose or suggest the step of determining the plurality of levels through the steps of: determining the first set by determining values of how many of each of the characters of the alphabet are in the previously selected portion; and determining the additional sets by assigning names for a given additional set to selected pairs of names from another of the sets, wherein each assigned name is unique to the names for a selected pair, as required by claims 2 and 21.

Claims 7 and 23

Claims 7 and 23 are rejected under 35 U.S.C. §102(b) as being anticipated by Floratos. In particular, the Examiner asserts that Floratos discloses to determine, for a name that has changed in the sets of names, a location in the input string that corresponds to the changed name (page 458, col. 1, line 5, to col. 2, line 32; FIG. 2).

In the text cited by the Examiner, Floratos discloses that,

> initially, when a query sequence Q is provided to the system, all $\boldsymbol{P} \in \boldsymbol{\Pi}$ that match Q are located. This can be done very rapidly by using a hashing variation of a technique presented in [20]. More specifically, for every position within Q we generate W hash values, one for every substring of length 2, 3, ..., $(W + 1)$ starting at that position. For every such substring, the corresponding hash value depends only on the first and last characters of the substring as well as on the number of residues between those two characters. Figure 1 provides an example of the process for a given query sequence.
>
> The hash entry corresponding to a particular value h contains all of the offsets p of the query sequence Q such that a substring (of length at

> most $W + 1$) starting at p hashes to the value h. For example, Figure 2 shows a snapshot of the hash table generated for a particular query sequence.
>
> To check whether a pattern $P \in \Pi$ matches Q, we use an array of counters $C\,[1.\,|Q|\,]$ of size equal to the length of Q. Initially, every entry of the array is set to 0. Starting at offset 1 in P, we locate all offsets j within P corresponding to a residue, excluding the offset corresponding to the last residue. For every such j, let R be the shortest substring of P starting at j and containing exactly two residues. Let OL denote the list of offsets in Q pointed to by the hash table entry corresponding to R. If OL is not empty, then for every offset $p \in OL$ the counter $C[p - j + 1]$ is incremented by one. If the pattern P contains exactly n residues, then at the end of this process the counter $C[i]$ will have the value $(n - 1)$ if and only if Q matches P at offset i. (An advantage of this matching technique is that it typically requires time which is sublinear to the size of the query sequence Q and depends only on the number of residues in the pattern P.) (Page 458, col. 1, line 6, to col. 2, line 30.)

Appellants could find no disclosure or suggestion in Floratos of determining for a name that has changed in the sets of names, a location in the input string that corresponds to the changed name. Claims 7 and 23 require the step of determining for a name that has changed in the sets of names, a location in the input string that corresponds to the changed name.

Thus, Floratos, Savitch, and Fredman, alone or in any combination, do not disclose or suggest the step of determining for a name that has changed in the sets of names, a location in the input string that corresponds to the changed name, as required by claims 7 and 23.

Claim 9 and 24

Claims 9 and 24 are rejected under 35 U.S.C. §102(b) as being anticipated by Floratos. In particular, the Examiner asserts that Floratos discloses wherein each of the names in the sets of names corresponds to a pattern, and further comprises the step of, when using changes in the names, to select permutation patterns from the patterns (page 458, col. 2, lines 16-32).

In the text cited by the Examiner, Floratos discloses that,

> initially, every entry of the array is set to 0. Starting at offset 1 in P, we locate all offsets j within P corresponding to a residue, excluding the offset corresponding to the last residue. For every such j, let R be the shortest substring of P starting at j and containing exactly two residues

> Let OL denote the list of offsets in Q pointed to by the hash table entry corresponding to R. If OL is not empty, then for every offset $p \in OL$ the counter $C[p - j + 1]$ is incremented by one. If the pattern P contains exactly n residues, then at the end of this process the counter $C[i]$ will have the value $(n - 1)$ if and only if Q matches P at offset i. (An advantage of this matching technique is that it typically requires time which is sublinear to the size of the query sequence Q and depends only on the number of residues in the pattern P.)
> (Page 458, col. 2, lines 16-30.)

Appellants could find no disclosure or suggestion in Floratos that each of the names in the sets of names corresponds to a pattern, and that the step of using changes further comprises the step of selecting permutation patterns from the patterns. Claims 9 and 24 require wherein each of the names in the sets of names corresponds to a pattern, and wherein the step of using changes further comprises the step of selecting permutation patterns from the patterns.

Thus, Floratos, Savitch, and Fredman, alone or in any combination, do not disclose or suggest wherein each of the names in the sets of names corresponds to a pattern, and wherein the step of using changes further comprises the step of selecting permutation patterns from the patterns, as required by claim 9 and 24.

Claim 10

Claim 10 is rejected under 35 U.S.C. §102(b) as being anticipated by Floratos. In particular, the Examiner asserts that Floratos discloses comparing names that have changed in the sets of names to a database comprising a plurality of stored names (page 458, col. 2, line 46, to page 459, col. 1, line 4).

In the text cited by the Examiner, Floratos discloses that

> an example of such a situation is depicted in Figure 3. In such cases, the individual segments corresponding to the *two* patterns must be *chained* into one. In particular, two segments (i, j, k, l), and (i', j, k', l') associated with S_j are designated as *compatible* iff $k \le k'$, $k + l + w\_len > k'$, and $k' - k = i' - i$, where w_len is an integer parameter (defined by the user) that allows for the chaining of segments which do not intersect as long as one begins no more than w_len positions after the end of the other. The segment resulting from chaining (i, j, k, l) and (i', j, k', l') together is $[i, j, k, \max(l, k' - k + l')]$.
> (Page 458, col. 2, line 46, to page 459, col. 1, line 4.)

Appellants could find no disclosure or suggestion in Floratos of comparing names that have *changed in the sets of names to a database comprising a plurality of stored names*. Claim 10 requires the step of comparing names that have changed in the sets of names to a database comprising a plurality of stored names.

Thus, Floratos, Savitch, and Fredman, alone or in any combination, do not disclose or suggest the step of comparing names that have changed in the sets of names to a database comprising a plurality of stored names, as required by claim 10.

Dependent Claims 2-19 and 21-28

Dependent claims 2-17 and 21-26 were rejected under 35 U.S.C. §102(b) as being anticipated by Floratos, claims 18 and 27 were rejected under 35 U.S.C. §103(a) as being unpatentable over Floratos, in view of Savitch, and claims 19 and 28 were rejected under 35 U.S.C. §103(a) as being unpatentable over Floratos, and Savitch, and further in view of Fredman.

Claims 2-19 and 21-28 are dependent on claims 1 and 20, respectively, and are therefore patentably distinguished over Floratos, Savitch, and Fredman (alone or in any combination) because of their dependency from independent claims 1 and 20 for the reasons set forth above, as well as other elements these claims add in combination to their base claim

Conclusion

The rejections of the cited claims under sections 102 and 103 in view of Floratos, Savitch, and Fredman, alone or in any combination, are therefore believed to be improper and should be withdrawn. The remaining rejected dependent claims are believed allowable for at least the reasons identified above with respect to the independent claims.

The attention of the Examiner and the Appeal Board to this matter is appreciated.

Respectfully,

Kevin M. Mason

Date: February 12, 2007

Kevin M. Mason
Attorney for Applicant(s)
Reg. No. 36,597
Ryan, Mason & Lewis, LLP
1300 Post Road, Suite 205
Fairfield, CT 06824
(203) 255-6560

APPENDIX

1. A method of discovering permutation patterns from an input string having a plurality of characters, each character being from an alphabet, the method comprising the steps of:

selecting a new portion of the input string, the new portion differing from a previously selected portion of the input string by at least one new character of the input string;

determining one or more values for how many of the at least one new characters are in the portion of the input string;

determining which, if any, names in a plurality of sets of names have changed by selection of the new portion, the plurality of sets comprising a first set and a plurality of additional sets, wherein the first set corresponds to all of the characters in the alphabet and to values of how many of the characters of the alphabet are in the previously selected portion, wherein the values are names for the first set, and wherein each additional set comprises names corresponding to selected pairs of names from a single other set; and

using changes in the names to determine the permutation patterns.

2. The method of claim 1, further comprising the step of determining the plurality of levels through the steps of:

determining the first set by determining values of how many of each of the characters of the alphabet are in the previously selected portion; and

determining the additional sets by assigning names for a given additional set to selected pairs of names from another of the sets, wherein each assigned name is unique to the names for a selected pair.

3 The method of claim 1, wherein the assigned names are codes.

4. The method of claim 3, wherein the codes are natural numbers.

5. The method of claim 1, wherein the step of determining which, if any, names in a plurality of sets of names have changed determines that a name has changed and further comprises the step of determining that a new name is needed for the changed name.

6. The method of claim 5, wherein the step of determining which, if any, names in a plurality of sets of names have changed further comprises the step of selecting a new name, not currently in use in the sets of names, for the changed name.

7. The method of claim 1, further comprising the step of determining for a name that has changed in the sets of names, a location in the input string that corresponds to the changed name.

8. The method of claim 7, wherein the changed name corresponds to at least two characters of the input string and a location in the input string of a given character of the at least two characters is chosen as the determined location.

9. The method of claim 1, wherein each of the names in the sets of names corresponds to a pattern, and wherein the step of using changes further comprises the step of selecting permutation patterns from the patterns.

10. The method of claim 1, further comprising the step of comparing names that have changed in the sets of names to a database comprising a plurality of stored names.

11. The method of claim 1, wherein the additional sets have names corresponding to only a single pair of names from another set.

12. The method of claim 1, wherein the step of using changes further comprises the step of correlating the changed names with permutation patterns.

13. The method of claim 12, wherein the step of determining which, if any, names in a plurality of sets of names further comprises, for each changed name, updating a count corresponding to that changed name, and wherein the method further comprises the step of:

performing the steps of selecting, determining one or more values, and determining which, if any, names in a plurality of sets of names until the entire input string has been selected.

14. The method of claim 13, wherein portions selected have a predetermined size, and wherein the method further comprises the step of selecting a number of predetermined sizes and performing the steps of selecting, determining one or more values, and determining which, if any, names in a plurality of sets of names for each of the predetermined sizes.

15. The method of claim 14, wherein the step of using changes further comprises the step of determining permutation patterns corresponding to counts greater than or equal to a predetermined count.

16. The method of claim 15, further comprising the step of determining maximal permutation patterns from the determined permutation patterns.

17. The method of claim 16, wherein the step of determining which, if any, names in a plurality of sets of names further comprises the step of determining location lists for each of the names corresponding to permutation patterns, and wherein the step of determining maximal permutation patterns further comprises the steps of comparing location lists for permutation patterns and eliminating duplicate permutation patterns by using the location lists.

18. The method of claim 1, wherein the at least one character is a single character and wherein the step of selecting further comprising selecting a portion

of the input string that differs from the previously selected portion of the input string by moving a window one character, from the previously selected portion, along the input string, the window selecting the new portion of the input string.

19. The method of claim 1, wherein the sets of names are stored in a balanced search tree.

20. An apparatus for discovering permutation patterns from an input string having a plurality of characters, each character being from an alphabet, the apparatus comprising:

a memory;

at least one processor coupled to the memory, the at least one processor configured:

to select a new portion of the input string, the new portion differing from a previously selected portion of the input string by at least one new character of the input string;

to determine one or more values for how many of the at least one new characters are in the portion of the input string;

to determine which, if any, names in a plurality of sets of names have changed by selection of the new portion, the plurality of sets comprising a first set and a plurality of additional sets, wherein the first set corresponds to all of the characters in the alphabet and to values of how many of the characters of the alphabet are in the previously selected portion, wherein the values are names for the first set, and wherein each additional set comprises names corresponding to selected pairs of names from a single other set; and

to use changes in the names to determine the permutation patterns.

21. The apparatus of claim 20, wherein the at least one processor is further configured, in order to determine the plurality of levels:

to determine the first set by determining values of how many of each of

the characters of the alphabet are in the previously selected portion; and

to determine the additional sets by assigning names for a given additional set to selected pairs of names from another of the sets, wherein each assigned name is unique to the names for a selected pair.

22. The apparatus of claim 20, wherein the at least one processor is further configured, when determining which, if any, names in a plurality of sets of names have changed determines that a name has changed to determine that a new name is needed for the changed name.

23. The apparatus of claim 20, wherein the at least one processor is further configured to determine, for a name that has changed in the sets of names, a location in the input string that corresponds to the changed name.

24. The apparatus of claim 20, wherein each of the names in the sets of names corresponds to a pattern, and wherein the at least one processor is further configured, when using changes in the names, to select permutation patterns from the patterns.

25. The apparatus of claim 20, wherein the additional sets have names corresponding to only a single pair of names from another set.

26. The apparatus of claim 20, wherein the at least one processor is further configured, when using changes in the names to determine permutation patterns, to correlate the changed names with permutation patterns.

27. The apparatus of claim 20, wherein the at least one character is a single character and wherein the at least one processor is further configured, when selecting a new portion of the input string, to select a portion of the input string that differs from the previously selected portion of the input string by moving a window one character, from the previously selected portion, along the input string, the window

selecting the new portion of the input string.

28. The apparatus of claim 20, wherein the sets of names are stored in a balanced search tree.

29. An article of manufacture for discovering permutation patterns from an input string having a plurality of characters, each character being from an alphabet, the article of manufacture comprising:

a computer readable medium containing one or more programs which when executed implement the steps of:

selecting a new portion of the input string, the new portion differing from a previously selected portion of the input string by at least one new character of the input string;

determining one or more values for how many of the at least one new characters are in the portion of the input string;

determining which, if any, names in a plurality of sets of names have changed by selection of the new portion, the plurality of sets comprising a first set and a plurality of additional sets, wherein the first set corresponds to all of the characters in the alphabet and to values of how many of the characters of the alphabet are in the previously selected portion, wherein the values are names for the first set, and wherein each additional set comprises names corresponding to selected pairs of names from a single other set; and

using changes in the names to determine the permutation patterns.

EVIDENCE APPENDIX

There is no evidence submitted pursuant to § 1.130, 1.131, or 1.132 or entered by the Examiner and relied upon by appellant.

RELATED PROCEEDINGS APPENDIX

There are no known decisions rendered by a court or the Board in any proceeding identified pursuant to paragraph (c)(1)(ii) of 37 CFR 41.37.